Jörg Reinholz
fastix WebDesign & Consult
Hafenstr. 67
34125 Kassel
☎ 0561 317 22 77
0561 217 22 76
http://www.fastix.org

Der UTF-8 Zeichensatz, ein Kurs für „Spezialisten“ und Webdesigner

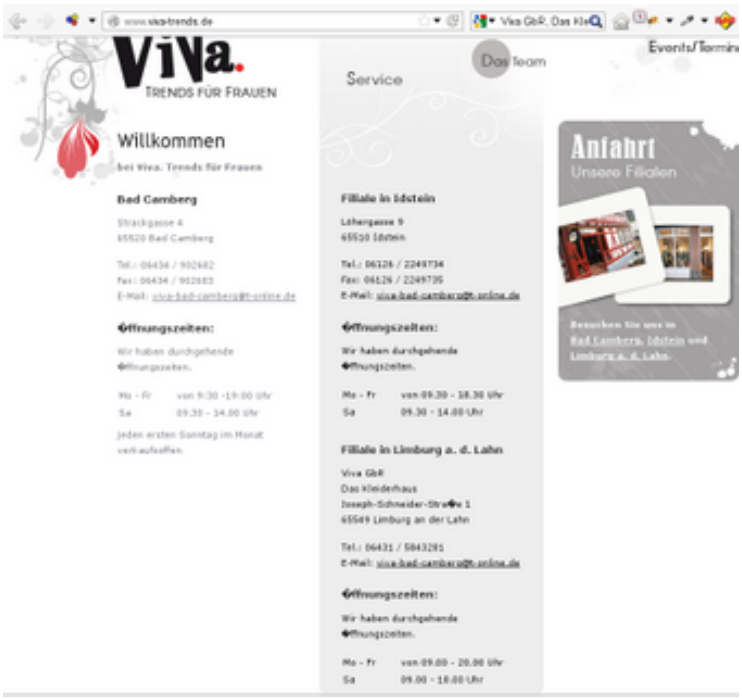
Die Begegnung mit dem "Zeichensatzteufel" ist nicht auf Webseiten kleiner Firmen beschränkt...

Denn eine große Düsseldorfer Firma (mit Ablegern in Berlin und Essen),

- die sich selbst mit kernigen Aussagen wie "Seit über 10 Jahren konzipieren und gestalten wir professionelle Internetauftritte für Unternehmen aus dem Mittelstand. Profitieren Sie von unserem Know how!" und "*Möglich macht dies unser einzigartiges Full-Service-Konzept sowie unsere langjährige Erfahrung im Web. Über 20.000 Unternehmen in Deutschland, Österreich und der Schweiz hat das Konzept bereits überzeugt. Eine professionelle Betreuung der Unternehmen stellen dabei die 640 Mitarbeiter an unseren verschiedenen Standorten sicher.*" *bewirbt*;
- die hunderte Außendienste (ca. 400 von 640 Mitarbeitern!) durch Republik schickt und im vollen Bewusstsein, auch hierbei zu lügen, erst telefonisch eine "*kostenlose Erstellung von Webseiten anbietet*" und
- dann im Vertragsgespräch psychologisch geschickt und mit weiteren Lügen tausenden durch diese und weitere arglistige Täuschungen den Betroffenen einen Vertrag aufschwatzt, auf dem diese über 4 Jahre insgesamt - wahnwitzig erscheinende und für die Leistung definitiv nicht angemessene - 7.000 bis 20.000 Euro verlangt (was diese Firma auch noch rotzfrech einklagt und worauf diese, wenn es denn mal gelingt, auch noch stolz sind)

... aber auch viele andere "Spezialisten" und "Webdesigner" haben ein Problem mit UTF-8. Das ist ein Zeichensatz mit dem sich z.B. deutsche Umlaute aber auch kyrillische, griechische arabische, russische, japanische und chinesische Zeichen darstellen lassen. Der setzt sich so nach und nach durch...

Nun, manchmal stößt man auf Webseiten, die so aussehen:



Diese Webseite wurde (neben hunderten oder gar tausenden mit dem gleichen Problem) von der oben beschriebenen Firma erstellt. Würde ich diese hier benennen, dann würde diese zweifelsfrei betrügerisch agierende Webdesign-Firma mich auch noch dreist verklagen.

Wie stellt man nun sicher, dass der UTF-8-Zeichensatz durchgehend und richtig angewendet wird? Dass eine Katastrophe wie die obige nicht passiert?

Hierbei ein Rezept auszustellen ist ziemlich schwierig, weil der Zeichensatz an vielen Stellen verbogen werden kann. Die erste Einstellung betrifft stets die zum Erstellen der Webseite verwendete Software. Will man UTF-8 senden, dann muss man natürlich auch UTF-8 eingeben. Das, so scheints, ist nicht jedem "Spezialist" oder "Webdesigner" so klar, wie es ihm klar sein sollte.

Verwendet man als Webserver einen Apache (das ist der meist genutzte), dann sollte man auch UTF-8 auch als Default-Zeichensatz angeben. Das geschieht in der Konfiguration des Apache Webservers selbst, des virtuellen Hostes oder in der Verzeichniskonfiguration. Und zwar mit:

```
AddDefaultCharset UTF-8
```

Häufig liest man, man solle im HTML-Header mit

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

den Zeichensatz einstellen. Doch das wirkt nur dann wenn der Server im HTTP-Header keine Angaben zum Zeichensatz sendet oder wenn die Date lokal, z.B. vom Dateisystem geöffnet wird. Das erklärt dann auch, warum die Datei vom Rechner geöffnet korrekt dargestellt wird, nach dem Hochladen und Abrufen vom Server aber die Umlaute beschädigt zu sein scheinen.

Arbeitet man mit PHP, dann sollte die php.ini folgende Einstellung enthalten:

```
default_charset = "utf-8"
```

Kommen Daten aus einer Datenbank, hier nehme ich mal MySQL als Beispiel, dann sollte die Datei "my.cnf" folgende Einstellungen enthalten:

Im Abschnitt [client]:

```
default-character-set=utf8
```

Im Abschnitt [mysqld]:

```
character-set-server=utf8
```

Natürlich sollten Tabellen und Spalten beim Anlegen auch entsprechende Einstellungen bekommen, denn das ist nur die Voreinstellung**für künftige Datenbanken**. Man muss darauf achten, dass man das nicht versehentlich überschreibt, wenn man eine Tabelle anlegt.

Etwas wie ein ....

```
CREATE TABLE `test` (
`text` TEXT NOT NULL CHARACTER SET utf8 COLLATE utf8_general_c
) ENGINE = InnoDB CHARACTER SET utf8 COLLATE utf8_general_ci;
```

... sorgt also für den richtigen Zeichensatz in der Tabelle sowie in den Spalten und die "richtige" Sortierung (DIN 5007-1)

Zudem kann man in PHP **nach einem Verbindungsaufbau zu MySQL**mit

```
mysql_set_charset ('UTF-8');
```

den Zeichensatz für die Verbindung zum mySQl-Server einstellen.**Ebenfalls nach einem Verbindungsaufbau** von PHP zur Datenbank kann man bei Bedarf immer noch noch folgende Anweisung senden:

```
mysql_query('SET NAMES=utf8');
```

Das macht die folgenden, auch möglichen Änderungen überflüssig:

```
mysql_query('SET CHARACTER_SET_SERVER=utf8');
mysql_query('SET CHARACTER_SET_CLIENT=utf8');
```

MultiByte-Funktionen

Weiterhin sollte man, so man mb-funktionen verwendet, noch

```
mb_internal_encoding("UTF-8");
```

verwenden um diese Funktionen auf den korrekten Zeichensatz vorzubereiten.

Alte Datenbestände konvertieren

generell lassen sich Datenbestände von ISO 8859-1 oder ISO 8859-12 oder CP 1252 (das wäre der Windows-Zeichensatz) konvertieren. Um eine Datei also vom alten Windows-Zeichensatz nach UTF-8 zu konvertieren wird sich etwas wie

```
$file='test.txt';
file_put_contents($file, iconv('Windows-1252', 'UTF-8//TRANSLIT', file_get_contents($file)));
```

als gangbarer Weg erweisen. Natürlich kann man das in eine Schleife packen, um alle Dateien eines Verzeichnisses neu zu kodieren.

Formulardaten

Formulardaten werden normalerweise in der Kodierung, also mit dem Zeichensatz zurück gesendet, in welchem der Browser das Formular empfangen hat (siehe oben). Man kann aber einem Formular ggf. mehrere - oder eben nur eine Kodierung mitteilen, welche der Server zu akteptieren bereit ist. Das ist wichtig, wenn der Server von dem das Formular kommt und der die

Daten empfangende Server nicht der selbe ist.

```
<form action="http://www.example.com/data.php" method="post" accept-charset="UTF-8">
```

Mail und UTF-8

Das zeige ich mal an einem Beispiel für PHP. Zuerst werden Sprache und die Kodierung für die Multibyte-Funktionen gesetzt. Dann werden (soweit diese Umlaute enthalten**können**) die Namen als Teil der Mailadressen kodiert - nicht jedoch das Subjekt. Beim Senden selbst darf man die header für die richtige Kodierung der Nachricht selbst und für den Transfer nicht vergessen:

```
mb_language("de");
mb_internal_encoding("UTF-8");

$to      = '"' . mb_encode_mimeheader('Jörg Reinholz') . '" <joerg.reinholz@fastix.org>';
$from    = '"Formmailer" <admin@fastix.org>';
$subjekt = 'Mit viel Hüh und Hott gehts flott.';
$message = 'Das ist eine Testnachricht mit Umlauten:
ÄÜÖ
äüöß
und dem Euro-Symbol:€';

$headers = array
(
     'Content-Type: text/plain; charset="UTF-8"',
     'Content-Transfer-Encoding: 8bit',
     'Reply-To: ' . $from
);
mb_send_mail($to, $subjekt, $message, implode("\r\n", $headers));
```

Falsche Umlaute in einer Konsole auf Unix- oder Linuxmaschinen?

Das sollte in den meisten Fällen mit einem

```
 ~> export LANG=de_DE.UTF-8
```

behoben sein. Eventuell in der Datei .profiles mit eintragen.

Falsche Umlaute bei der Ausführung von Systembefehlen?

Das gleiche Rezept! Hier mit der PHP-Funktion "exec" und dem Shell-Befehl "ls -al" als Beispiel:

```
exec("export LANG=de_DE.UTF-8 && ls -al");
```

Falsche Umlaute in einer Konsole bei einem Verbindungsaufbau mit Putty

Das habe ich in diesem Artikel beschrieben.

Weitere Informationen in diesem Bereich:

- Realisierter Kundenwunsch: Perfekte Rechnungen online erstellen
- Die fastix Netztools (whois, ping, traceroute, dig, nmap)
- osCommerce 2.3.1 (banner_manager.php) repariert
- Wordpress(-Plugin) subster-rejuvenation repariert
- Ticketsystem für mittelständischen Webshop-Betreiber fertiggestellt
- Joomla 1.6 und der Meta-Tag

- htpasswd-Generator - Eine Lösung für kleine Webs
- Franchisingfähiger Webshop fertiggestellt